Shape representation and coding of visual objets in multimedia applications – an overview

Corinne LE BUHAN JORDAN *
Sushil BHATTACHARJEE *
Frank BOSSEN *
Frédéric JORDAN *
Touradj EBRAHIMI*

Abstract

Emerging multimedia applications have created the need for new functionalities in digital communications. Whereas existing compression standards only deal with the audio-visual scene at a frame level, it is now necessary to handle individual objects separately, thus allowing scalable transmission as well as interactive scene recomposition by the receiver. The future MPEG-4 standard aims at providing compression tools addressing these functionalities. Unlike existing frame-based standards, the corresponding coding schemes need to encode shape information explicitly. This paper reviews existing solutions to the problem of shape representation and coding. Region and contour coding techniques are presented and their performance is discussed, considering coding efficiency and rate-distortion control capability, as well as flexibility to application requirements such as progressive transmission, low-delay coding, and error robustness.

Key words : Rewiew, Multimedia sercice, Image coding, Geometrical shape, Edge detection, Information compression, Object oriented method, Standardization, Intraframe coding, Interframe coding.

REVUE DES MÉTHODES DE REPRÉSENTATION ET DE CODAGE DE FORMES D'OBJETS VISUELS DANS LES APPLICATIONS MULTIMÉDIA

Résumé

Les besoins en matière de fonctionalité orientées objet dans les communications audiovidueIles sont apparus récemment avec l'émergence d'application nouvelles telles que la video conférence, les vidéophones et la vidéo interactive. Alors que les normes de compression existantes traitent la scène audio-visuelle au niveau de la trame, il est maintenant nécessaire de traiter séparément les differents objet présents, permettant ainsi une transmission échelonnable aussi bien que la recomposition de la scène par le receveur. La future norme MPEG-4 a pour but de proposer des outils de compression offrant ces nouvelles fonctionalités. Contrairement aux standards orientés trame existants, les schémas de codage correspondants doivent intégrer l'information de forme. Cet article présente un certain nombre de solutions existantes au problème de la représentation et du codage des formes. Différentes techniques de codage de formes et de contours sont présentées et leurs performances sont analysées en considérant l'efficacité du codage et la capacité de régulation débit/distortion, ainsi que la flexibilité vis-à-vis des besoins de l'application, tels que la transmission progressive, le codage à court délai, et la résistance aux erreurs.

Mots clés : Article de synthèse, Service multimedia, Codage image, Forme géometrique, Détection bord, Compression information, Méthode orientée objet, Normalisation, Codage intratrame, Codage intertrame.

Contents

I. INTRODUCTION

The progress in microelectronics and computer technology coupled with the creation of networks with various channel capacities is the basis of an infrastructure for a new era of data processing. Emerging applications such as video conferencing, mobile videophones and multimedia will have a great impact on professional life, education and entertainment.

The digital representation of visual information in its canonical form leads to a huge amount of data. Early work in visual data compression attempted to reduce the amount

* Signal Processing Laboratory, Swiss Federal Institute of Technology at Lausanne - EPFL, CH - 1015 Lausanne, Suisse

of bits required to present such data, while preserving the original visual quality as much as possible. Besides transform based methods which use tools for energy compaction attemped to achieve this goal object oriented schemes were developed in order to mimic the processing of the human visual system.

More recently, due to the developments in multimedia applications (editing, video games and interactive video, computer generated graphics, etc.) new functionalities such as object manipulation, selective object rendering, different temporal resolution of objects, and so on, are as important as improved compression efficiency. Object oriented coding seems a more natural approach to practical systems as in these applications, the original source material is frequently composed of different objects put together in a mosaic form or in a layered fashion.

A major difference between conventional techniques based on energy compaction (such as the well known block based DCT method) and object oriented methods is that in addition to texture (and motion in video), information about the shape of every object has to be encoded and communicated to eventual decoders. Hence, shape coding is a topic of interest in all object oriented techniques. The MPEG-4 standardization activity for multimedia applications has examined various shape coding techniques for object oriented coding [36].

In this paper, we discuss major shape coding techniques described in literature and review their efficiency for coding of typical objects. The structure of the paper is as follows. Section II outlines the problems tackled by the various shape representation methods used in litterature The major classes of shape coding techniques are also briefly overviewed in this section. Section III describes bitmap-based coding methods, in which shape information is processed as a binary image. Section IV presents two techniques based on an intrinsic shape coding principle, namely, a skeleton-based and a quadtree-based method. Section V describes a popular techniques for shape coding known as 'chain coding' which is based on a contour representation. Variants of the same technique are also described in the same section. By relaxing the losslessness constaint, geometrical representation methods are alternatives to chain coding which are also based on contour repesentation but lead to a higher performance in terms of compression efficiency. Section VI presents a number of such geometrical representation methods.

In video coding applications, the temporal correlation between successive instances of the same object may be exploited to enhance the compression efficiency. These techniques are described in section VII. As it is the case in conventional image and video coding, there is no objective measure in order to assess the perceived visual quality of a lossy coded shape. In addition to this problem, the definition of the rate when coding an object in a scene is less obvious when compared to conventional image and video compression. These issues are discussed in section VIII. In section IX, the performance of some of the techniques described in this paper is studied. Conclusions are drawn in section X.

II. OBJECT SHAPE REPRESENTATION

The problem of region shape representation has been investigated in the past for arbitrary and constrained regions [3,17]. The different representations are closely linked to three major classes of shape representation techniques, namely bitmap, intrinsic and contour based ones. Designed only for compression, bitmap-based techniques apply binary image coding methods, such as those developed for facsimile transmission, to shape images. Intrinsic shape representation methods either decompose the shape into smaller, simpler elements (quadtree-based techniques [17], fractal representation, etc.), or represent it by its skeleton.

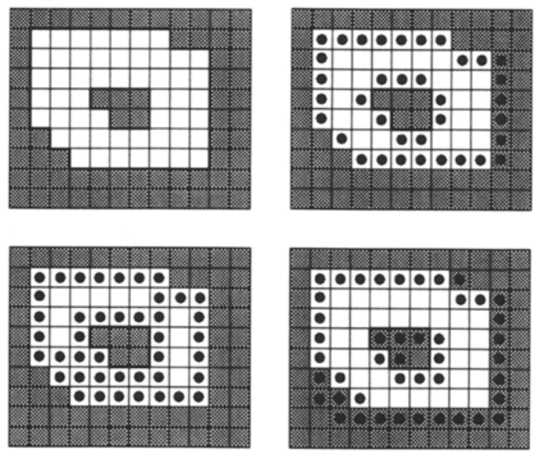

FIG. 1.- Different contour representations : (upper-left) 6-connected with crack edges ; (upper-right) 8-connected ; (lower-left) 4-connected ; (lower-right) alternative 4-connected with non-redundant inter-region representation.

Différentes représentations des contours : (en haut à gauche) 6-connectivité avec « crack edges » ; (en haut à droite) 8-connectivité ; (en bas à gauche) ; 4-connectivité (en bas à droite) représentation en 4-connectivité sans redondance inter-régions.

As opposed to the latter region-based methods, contour-based techniques use a transform to convert the object masks into contours, like in the human visual system. This representation is desirable in applications where a semantic or geometrical description of the shape is used, such as in database applications [34]. For rendering, an inverse transform is used to recover the shape. This transformation should preserve information to allow lossless shape coding using contours. When simplification hypotheses are available, for example if every region is at least two pixels wide, the transformation can often be expressed in a very compact form. Various contour transformations are depicted in Figure 1. There are several connectivity schemes, which define what relative positions two neighboring contour pixels can take. The two most common schemes are 4-connectivity and 8-connectivity. Let i and j be two neighboring contour pixels. In a 4-connected contour, pixel i is either to the north, east, south, or west of pixel j, and in a 8-connected contour it can also be to the north-east, south-east, south-west, or north-west in addition to the previous four directions. Let **O** be the object mask defined as

(1) $$O_{xy} = \begin{cases} 1 \text{ if the pixel located at } (x, y) \text{ belongs to the} \\ \text{object, 0 otherwise.} \end{cases}$$

Let N_{xy} be the set of pixels neighboring the pixel located at (x, y). The contour mask C (N) is defined as

(2) $$O_{xy} = \begin{cases} 1 \text{ if } O_{xy} = 1 \text{ and } \exists(u, v) \in N_{xy} \text{ such that} \\ O_{uv} = 0, 0 \text{ otherwise.} \end{cases}$$

For 4-connected contours $N_{xy} =$

$$\{ (u, v) \mid \max (|u - x|, |v - y| = 1\}$$

and for 8-connected contours $N_{xy} =$

$$\{(u, v) | (|u - x| + |v - y| = 1\}.$$

An alternate connectivity scheme is based on 6-connected contour points. Whereas 4- and 8- connected contour points lie on pixels, 6-connected contour points lie *between* pixels (crack edges). Although it facilitates handling of thin shape details such as isolated pixels or lines [42], it does not seem to be as popular as 4- and 8- connectivity schemes.

III. BITMAP CODING

Bitmap coding techniques operate on the raw data, without performing any prior transformation. Two such techniques are presented in this section. Both are based on paradigms that have been successfully applied to black and white image coding, and for facsimile applications in particular. The first one is based on run-length encoding and the second on arithmetic coding with conditional probabilities.

III.1. Modified-modified read (MMR)

The MMR method is based on run-length encoding of a binary image. The image is scanned line by line and the lengths of each black and each white segment are encoded. To improve the efficiency of the method the encoding of the length is optimized by taking into account segment boundaries in the previous line. This algorithm has been successfully applied in the facsimile group 4 standard [25].

An adaptation of this method to shape coding is proposed in [46]. Macroblock partitioning is introduced for compatibility with DCT-based texture coding. Furthermore a size conversion procedure is proposed to achieve lossy shape coding.

III.2. Context-based arithmetic encoding (CAE)

Arithmetic coding [45] is a very efficient entropy coding scheme. Given the probability distribution of symbols in a string, arithmetic encoding can compress the string to a length close to its entropy. The standard application of arithmetic encoding to a binary image is as follows. Pixels of the image are encoded in a predefined scan order, typically raster scan. At each pixel to be coded, the arithmetic encoder is fed the value of the pixel plus a probability distribution. In the case of a binary image, this probability distribution reduces to a single number p_0 which gives the probability of any pixel being *black*. The probability of any pixel being *white* is simply given by $1-p_0$.

The efficiency of such a representation depends upon the suitability of the probability distrubution that is used. CAE assumes that a high degree of local correlation exists in the image. Hence, conditional probabilities are introduced. For a given pixel, the probability distribution is conditioned upon the values of the pixels in a local neighborhood [29]. The shape and size of the neighborhood is reprensented by a template (see Fig. 2 for the template used for *intra* mode coding). The size of the template is typically 10 pixels leading to 1024 different *contexts* [5]. A context is an integer specifying the value of each pixel within the template. The context is used to access a table containing probability distributions (one for each context). This table is created by a training procedure prior to coding, or in the case of adaptive CAE, it is adapted during the coding procedure as in JBIG [26]. Adaptive CAE is not considered here due to the small size of the images to be coded, which do not allow enough time for proper adaptation.

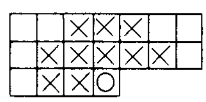

FIG. 2. – 10-pel template used for *intra* mode coding. The circle represents the current pixel to be coded and the crosses the pixels belonging to the template.

Voisinage de 10 éléments utilisé pour le codage intra. *Le cercle représente l'élement courant à coder et les croix les éléments appartenant au voisinage.*

The efficiency of CAE schemes largely depends on the choice of the template. In general, the larger the template, the better, the performance. However, the memory for storage of the probability tables increases exponentially with the size of the template. There is also the risk of over-generalization when using large templates.

In a block-based texture coding environment, the CAE method can be adapted to proceed in a block-based fashion [6].

IV. INTRINSIC SHAPE CODING

Intrinsic shape representation considers the region rather than its boundaries. Such representation methods, also called internal representation methods, were among the first methods to be investigated in the fields of computer vision and pattern recognition [3]. In this section, two techniques are presented. They are based on two different approaches to the problem: the first one consists in representing a shape by its skeleton, and the second one is a hybrid block/quadtree decomposition which has been investigated in the framework of the MPEG-4 video compression standardization process.

IV.1. Skeleton decomposition

Skeleton decomposition, also referred to as medial axis transform (MAT) or symmetric axis transform (SAT), has been widely investigated. It consists in describing a full figure by its skeleton. Basically, the skeleton of a shape can be extracted by taking the centres of all maximally inscribed discs. The shape description consists of two parts: the symmetric axis or locus of the centers of these discs and their corresponding radii towards the shape boundary (quench values). Mathematical morphology operators can be used to extract the skeleton and corresponding quench values for any pixel-quantized structural elements (disk, square, cross, etc.) [43]. Since different structural element result in different shape representations, the element used must be either implicitly known by the decoder or transmitted together with the skeleton and quench values. Shape representation parameters can be separately encoded by means of run-length coding combined with adaptive arithmetic coding [8].

The cost for transmitting all skeleton points remains too high for practical applications. It is possible to decrease the associated redundancy by introducing a minimal skeleton, which allows the use of disconnected skeleton points by removing redundant points [33]. Further improvement can be achieved by optimizing the structural element for the shape to be coded. Optimization can be performed at a reasonable cost by means of genetic algorithms [8, 9, 31]. Either the number of skeleton points or the final bitrate after entropy coding may be used as a cost function. It is also possible to maximize the area of the shape resulting from the reconstruction by only a few skeleton points. Experiments show a gain of up to 20% in final compression, using such techniques [8].

Skeleton-based shape decomposition is lossless and allows progressive transmission: partial reconstruction is possible from a subset of the transmitted information. Skeleton points with large quench values are more meaningful than those with smaller values, because they cover a larger image area. Therefore skeleton points should be transmitted in the decreasing order of their quench values. In practice, the bitstream is divided into different levels of priority, each one corresponding to a subset of quench values. Since the human visual system is sensitive to shape changes as small as one pixel, the method should assign to each progressive level one specific quench value [1, 2, ...,k] and all remaining ($> k$) values to the coarsest level. Entropy coding of the skeleton points is then performed separately for each level, which causes progressive transmission to be more expensive than non-progressive transmission. On the other hand, with the proposed quench thresholding, quench information does not need to be transmitted for refinement levels. Experiments on the binary image *girl* using 7 refinement levels have shown a moderate increase of about 10% in bitrate compared to non-progressive transmission [8].

The method is also stable with regard to transmission errors, since the loss of one skeleton point will only locally influence the shape which can still be reconstructed.

IV.2. Hybrid macroblock / quadtree for binary shape representation

The shape coding method described in [16] is based on a 16 × 16 pixel macroblock partition to facilitate its integration in existing block-based video coding schemes. A block is said to be *uniform inside* if all the pixels in it belong to the region coded, and *uniform outside* if all do not. These blocks are simply indicated to the decoder through a simple flag. Each non-uniform block is further recursively split into four square subblocks. The resulting block hierarchy is represented in a quadtree. Uniform outside and uniform inside subblocks are tagged as leaves, while non-uniform subblocks are further split. The resulting combinations of outside leaves, inside leaves and split subblocks are described by indices that are entropy coded with Huffman variable length code-words (VLC). In addition, index swapping procedures have been introduced to increase coding efficiency [16].

Although this technique is instrinsically lossless, filtering the shape before coding can increase compression performance and facilitate rate control. A preprocessing stage is applied to increase the proportion of large uniform blocks which require less bits for coding. One drawback of this approach is the introduction of blocking artifacts along the shape contours. However, these may be reduced by appropriate post-filtering.

V. CHAIN CODING OF CONTOURS

As mentioned in section II, regions may also be represented by their contours. The contour of a region is a connected chain of pixels. In this section, we discuss methods for representing contours by chains of adjacent boundary pixels. Both lossless and lossy representations are discussed.

V.1. Lossless coding

Freeman's *chain code* [18] may be used to represent contours of regions efficiently. Chain coding is based on the observation that the relative positions of any two adjacent points on a line in a digital image, are restricted to one of eight configurations (four possible configurations if 4-connectivity is considered). Thus, from a given point on the line, the transition to the next point can occur in one of eight directions. Each transition from one point to the next is called a *link* . The chain code for a line is formed by a sequence of links. The chain code representation of a contour is constructed as follows.

1. Select a starting point on the contour. Represent this point by its absolute coordinates in the image.
2. Represent every consecutive point by a link showing the transition needed to go from the current point to the next point on the contour.
3. Stop if the next point is the initial point, or the end of an open contour. In the case of open contours, either a special end symbol may be encoded or the total number of chain symbols may be transmitted prior to the symbols.

Such a chain code is referred to as *Freeman code.* Since every pixel in the contour is represented in the chain code, this method is lossless for encoding contours. Below, we discuss the basic Freeman code and some other schemes derived from it.

V. 1. 1. Freeman codes

Freeman codes have been widely accepted as the preferred method for representing contours, as they are highly efficient compared to the naive representation where each point on the countour is represented by a pair of absolute (integer) coordinates. Since a link in a Freeman code represents one of only eight possible directions, a set of eight symbols is sufficient to represent a Freeman-coded contour. The eight directions for the valid transitions may be named *East, North-East, North-West, West, South-West, South, and South-East.* Each of the eight symbols may be represented by a string of 3 bits. Thus, the contour may be encoded in a binary bit-string with 3 bits per link.

The above Freeman coding scheme is based on the 8-connectivity of contour points. Alternatively, we may choose to represent a contour using 4-connectivity, where only transitions in four directions, namely, *East, North, West,* and *South* are allowed. Thus, a set of four symbols is sufficient to represent contours based on 4-connectivity. This results in a Freeman code costing only 2 bits per link. Ostensibly, this gives us an improvement of 33% in coding efficiency. However, empirical results show that Freeman codes using 4-connectivity are, on average, 33% longer than the corresponding 8-connected chains [41]. In fact, 4-connected chains are not efficient to represent diagonal moves, requiring two links where 8-connected chains only use one link. As the geometric resolution of the image increases, 8-connectivity becomes more and more efficient compared to 4-connectivity. Hence, 4-connectivity is not commonly used for constructing Freeman codes for contours.

V.1.2. Differential coding

Differential chain codes, also proposed by Freeman [19], can offer substantial improvement in encoding efficiency, compared to Freeman codes. Our discussion of differential chain codes is based on 8-connectivity. The differential chain code representation for a contour may be constructed based on its Freeman code. Here, each link is replaced by its difference from the preceding link. Mathematically, the Freeman code C of a contour may be considered as an ordered-set of links:

$$C = \{ l_i \} \quad i = 1, 2, \ldots N$$

where l_i represents the i-th link in the chain of length N. Let $k_i = l_i - l_{i-1}$, for $i = 2, 3, \ldots N$. We can now define:

$$d_i = \begin{cases} k_i + 8 \text{ if } k_i < -3 \\ k_i - 8 \text{ if } k_i > 4 \\ k_i \quad \text{otherwise} \end{cases}$$

where l_i is the i-th link in the differential chain code. (The value of d_i is computed modulo 8 for 8-connected chains. Equivalently, the links for the differential chain code corresponding to a 4-connected Freeman code are computed modulo 4.)

Entropy coding may be applied to encode differential chain codes efficiently. The number of symbols required to represent a differentially coded chain is the same as that required for the corresponding Freeman code. However, unlike in the case of Freeman coding, the distribution of the symbols in the differential chain code is not uniform. We can expect the directions represented by two consecutive links in the Freeman code to be similar. Therefore, the difference between the links is generally small. For images with sufficiently fine resolution, most links in the differential chain code have values of −1, 0, or 1. A suitable entropy based encoding scheme may be developed to exploit this knowledge. A simple Huffman coding scheme can be used, associating a variable length code to different possible symbols. The most likely differential value, 0, will be assigned a single bit codeword and so on [24]. In general, a rate of about 2 bits per link may be expected by using Huffman encoding of differential chain coded contours. Result for this technique are presented in section IX.

V.1.3. Improved entropy coding

Several attempts to improve the efficiency of differential chain codes have been reported in the relevant literature. These methods usually rely on specific knowledge about the characteristics of the contours (or lines) in question, in order to establish thighter upper-bounds on the entropy of the corresponding differential chain codes. This leads to improved efficiency in entropy-based encoding.

Kaneko and Okudaira [27] proposed an algorithm for coding smooth contours, in which the contour is first divided into a sequence of segments. Within a segment, each link must be in one of the two adjacent directions, thus only requiring one bit per link. The length and the dominant direction of the segment must also be transmitted. The smoother the contour, the longer the segments, and hence, the higher the encoding efficiency. For the case of rough contours, the average length of each segment is small. In such cases this method offers about the same efficiency as the method described in section V. 1.2. For example, for a contour where the average segment-length is 3.4 links, the authors quote a coding rate of 2.02 bits per link. However, in another example, where the average segment-length comes to 16.5 links, the coding rate is 1.38 bit per link.

Eden and Kocher [15] have proposed an encoding scheme in the context of 4-connected Freeman coded contours. Their method is based on a first-order analysis of 4-connected contours. They introduce three symbols to represent different combinations of pairs of adjacent links : *left turn*, and *straight ahead*. These three symbols are used to represent the 4-connected contour. The resulting entropy for a contour with N points is $N \log_2 (3)$. Note that, expect in the case of very short contours, a right turn is never followed immediately by another right turn. The same can be said for left turns. Using the constraint that consecutive turns of the same type should never occur, Eden and Kocher reduce the bitrate per link from $\log_2 (3)$ down to $\log_2 (1 + \sqrt{2})$.

Lu and Dunham [32] use Markov models to describe the structure of the chains. The differential chain code is considered as a Markov source, and is encoded using an entropy based encoding scheme. The transition probabilities pertinent to the chosen Markov model are computed experimentally. These probability values are used to construct appropriate entropy based encoders. Using a Huffman encoding scheme based on a second-order Markov process, they achieve encoding rates similar to those obtained using the method of Kaneko and Okudaira, described above. However, the Markov model based scheme is simpler to implement. They show that significantly better encoding rates can be obtained by using an arithmetic encoding scheme [40] based on a second-order Markov model. On average, this method performs about 25% better than the simple Huffman encoding scheme for differential chain codes.

Extending the latter approach to adaptive, higher order Markov models, chain coded contours can be efficiently compressed using *prediction by partial matching* (PPM) [4, 11]. PPM is a finite-context statistical modeling technique (also referred to as a fixed-order, finite-context Markov model), that blends together several fixed-order context models to predict the next link in the contour. Prediction probabilities are computed from frequency counts, which are updated as links are processed. The link that actually occurs is encoded relative to its predicted distribution, using arithmetic coding. Although predictions generally get more accurate as more links are processed, they are not guaranteed to do so. This technique can be applied to either Freeman or differential codes. In the results shown in section IX, the former have been used, since it has been empirically found that they yield higher efficiency. Markov-modeled arithmetic coding methods perform well mainly because the probability estimates are very well suited to the data at hand. However, the additional cost of storing, and eventually maintaining the transition tables must also be taken into account.

V.2. Lossy schemes

The chain coding methods discussed in section V. 1. are lossless, as the encoded contour can be reconstructed exactly. At the expense of accuracy, some encoding efficiency may be gained by using a lossy chain coding method or by pre-processing the shape to make its contours smoother and consequently facilitate its chain coding.

A lossy chain coding method called *multi-grid chain code* (MMC) has been recently proposed by Salembier, Marqués and Gasull [24,42]. The MMC is based on the so called *hexagonal grid*, or 6-connected crack edge structure as mentioned in section II. By exploiting its specific grid structure, the technique gathers consecutive transitions together and exploits the corresponding introduced decoding ambiguity to yield a slightly lossy representation. According to the authors, the loss is negligible and the performance is better than the simplified 4-connected Freeman differential coding (three symbols).

Another method for producing a compact chain code description consists in simplifying the chain code so that only one pixel forward moves are allowed [24]. This technique requires 8-connectivity to be used and is not suited to represent shapes with high curvature. It yields a subsequent simplification since only three symbols are needed to represent the corresponding differential code in 8-connectivity.

One way to improve the performance of lossless chain coding schemes without adapting the corresponding algorithms consists in pre-processing the shape.

Smooth contours are indeed cheaper to encode. Moreover, a pre-filtering step is particularly relevant for noisy masks obtained from an imperfect former segmentation stage. Various filters may be used, such as a median filter, a morphological open-close-close-open filter, or the majority filter introduced in [22].

VI. GEOMETRICAL REPRESENTATION METHODS

Shape boundaries can be approximated by a 2-dimensional representation. Compression is achieved by transmitting only a set of representative vertices, which are then used to reconstruct the contour via line or spline segments. Geometrical approximation methods are usually lossy, and allow simple distortion control by constraining the geometrical representation accuracy to a maximum allowed error. Several geometrical representation methods exist based on different vertex selection, curve reconstruction and entropy coding approaches. Whereas entropy coding constitutes a relatively independent problem, vertex selection and curve reconstruction are closely linked, vertices being optimally selected with regard to a specific geometrical reconstruction model.

VI.1. Polygonal approximation

Polygonal approximation is the simplest geometrical approximation, where only the positions of the vertices need to be encoded. Since digital images are sampled on a finite grid, a lossless representation can be achieved for any shape, corresponding to the removal of intermediate contour pixels in contour straight lines : in the worst case where no straight lines exist in the shape boundaries, all contour pixels are polygon vertices. The ploygonal approximation encoding process consists of contour extraction, vertex selection and vertex coding. The decoding process requires vertex decoding followed by a polygon filling procedure [17].

In the context of data compression, the polygonal approximation which has the least number of vertices, and which conforms to a predefined error criterion, is generally considered as the optimum. Different error criteria may be used to measure the approximation quality [39]. A simple, sub-optimal, vertex selection algorithm based on iterative refinement was originally proposed in [13, 37]. The first approximation is given by the main axis of the shape, which is the longest segment between any two contour pixels. This approximation is recursively refined by inserting a new vertex on the contour point which is furthest away from the current approximation. This vertex insertion procedure stops when the distance between the approximated contour and any contour pixel is smaller than a given threshold (Fig. 3). More sophisticated algorithms, which do not only split but also merge segments, have been developed [1, 38]. A review of polygonal approximation methods is given in [14].

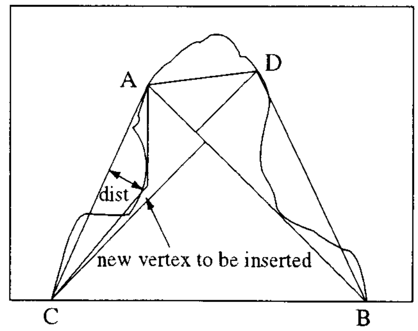

Fig 3. Iterative vertex selection.
Sélection itérative des sommets.

Efficient schemes encode only the position of the first vertex in absolute coordinates. The subsequent vertices are differentially coded, relative to the previous vertex. To actually improve compression, entropy coding is used. When the maximum allowed error is large, only a few vertices are needed. Their positions are usually not highly correlated, and entropy coding does not improve compression performance. When the maximum error is small, the vertices are numerous, and close to their neighbors. Entropy coding then becomes more efficient. In the limit, when the error is set to zero, polygonal approximation becomes very much like chain coding, and corresponding entropy coding schemes may be used.

VI. 2. Higher order approximations

A higher order geometrical representation can be provided by cubic curves interpolation. To reconstruct the approximated curve between two vertices, two constraints, such as derivatives or control points, are considered in addition to the coordinates of each endpoint. The resulting approximation is smoother and is visually more pleasant. However, it is not well suited to represent sharp angles.

VI. 2. 1 Higher order approximation schemes

A contour coding method based on approximation by blending curves was proposed in [44] to achieve rate-distortion optimization for a region-based coding scheme. The number of vertices is constrained by the available rate budget. Vertex positions are calculated in order to minimize the approximation distortion. Any distortion measure may be used. A popular one is the maximum local error defined as the distance between the original contour and the spline. Initially, the corresponding vertices are equidistantly distributed along the contour to be approximated. A steepest descent optimization technique, consisting in successively perturbing each vertex along the eight possible directions, is then applied. For each possible move, the four splines affected by the perturbed vertex are updated, and the corresponding error is computed. The position minimizing this error is retained. The algorithm is iterated until the error drops below a predefined threshold.

The algorithm minimizes the error for only one vertex at a time and the final solution is generally suboptimal. Moreover, whereas the algorithm requires a predefined number of vertices, no method is proposed to adapt the number of vertices to shape features. In [44] the contour optimization is carried out for different numbers of vertices and the solution minimizing the rate-distortion criterion is chosen (rate and distortion being defined in that case with regards to the whole region-based video coding scheme, not only the shape coding module).

Another method uses Hermite curves [17] to approximate the contour [21]. At each vertex the tangent of the reconstructed curve is constrained by the tangent of the original contour. Vertices are recursively selected according to a scheme similar to the ones used for polygonal approximation. As opposed to the former method that constrains the rate, this method constrains the distortion.

VI. 2. 2. Hybrid polygonal / cubic representation

Polygonal approximation is not suitable for smooth shapes, whereas cubic curve approximation cannot represent sharp edges. Therefore, a hybrid polygonal/cubic approximation is expected to be more efficient [20]. Basically, two approaches are possible: either the polygon / curve switch and the encoding of the corresponding one bit flag are done at vertex level, or at region level, relying on the assumption that object shapes are likely to be best described either by smooth curves only or distinct angles only, not by a combination of the two. With regard to the curve or polygon only approximations, the latter scheme requires only one additional bit for each region while exploiting the advantages of both techniques. Moreover, both schemes require the implementation of a decision scheme, either at a global or at a local level. In practice, both approximations are considered and the better one is used. However, this raises dependency problems when a local switch is to be applied, since spline and polygonal approximation methods do not share the same vertex selection procedure.

VI.2.3. Joint block / Hermite curve representation

The novel method proposed here is based on a 8×8 pixel block partition and approximates contours using Hermite curves. Only *control blocks* contain information to reconstruct the curve, by defining a vertex and a tangent vector using a line segment. For each control block, this line segment is defined to be the one which matches most accurately the contour within the block.

A greedy algorithm is used to select the control blocks. It starts with a pair of fixed control blocks taken from the list of all possible control blocks, that is, all blocks which contain at least one contour pixel. New control blocks are iteratively added until one of the following conditions is met: (i) the number of control blocks has reached a predefined maximum; (ii) the total error is lower than a given threshold; (iii) the insertion of new control blocks does not decrease the error anymore. At each iteration, the newly inserted control block is the one that maximizes the error decrease, among block candidates.

When reconstructing the curve, the norm of the tangent vector is scaled between the two endpoints. If two successive control vertices are far apart, the tangent constraint is tight, whereas if the vertices are close, the constraint is loose. The maximum curvature of the reconstructed contour can therefore be controlled roughly.

For efficient coding, the following algorithm is used. First the number of possible line segments within a control block is reduced using vector quantization [35]. The resulting codebook contains 16 entries, and thus 4 bits are needed to encode the line segment. Then the position of a control block is coded relatively to its predecessor. A small number of bits is used to do so. If two blocks are too far apart, an escape code is used, and the absolute position of the next block transmitted. Further compression gains may be achieved by the use of entropy coding.

The accuracy of this method is limited by the size of the bothblock and the codebook. However, some modifications can be undertaken to enhance the quality of the reconstruction. One possibility is to allow two control vertices and tangent vectors in a block, and another is to increase the polynomial degree of the reconstructed curve. Similarly a method for representing sharp angles could be used to permit quasi-lossless or lossless re-construction. Also, the control block selection process can be improved by using more powerful optimization techniques such as genetic algorithms, but at the cost of speed.

To a certain extent, this technique enables progressive transmission. First only a carefully chosen subset of all control blocks is transmitted, and then each new transmitted block refines the contour.

VII. TEMPORAL PREDICTION

Shape coding techniques presented so far do not take advantage of temporal redundancies in the scene. In the framework of video sequence coding, it is important to also reduce temporal redundancy in the bitstream, shapes remaining very similar from frame to frame.

The potential drawback of any temporal prediction scheme is that to decode the k-th frame, the $k-1$ previous ones need to be decoded first. A solution to this problem is to code every f-th frame without temporal references. According to the needs of the application, a tradeoff must be found between high compression ratios (large f) and fast random access (small f).

VII.1. Motion based prediction schemes

Similar to texture prediction, shape prediction from frame to frame can be achieved by means of motion estimation and compensation techniques. A general motion model would handle shape rotation, scaling and projection. Several shape representation methods invariant to scaling and rotation have already been investigated [38]. If such a method is applied for intra-frame shape coding, only motion model parameters need to be transmitted to predict the shape in the next frame.

In practice however, when shape coding is performed in a complete shape-motion-texture coder, a scheme consistent with that of texture coding may be used to simplify the overall algorithm. The main difficulty lies in the dependency between shape, motion and texture coding. Motion and texture coding are expected to be more efficient when using shape data. This is the idea behind second generation coding techniques [28]. Temporal shape coding has been investigated mainly in the framework of region-based coders, where a spatio-temporal segmentation and tracking algorithm extracts a set of regions coherent both spatially and temporally [2,8]. In this case, each region is tracked along the sequence, and can therefore be temporally predicted from its previous instance. Shape prediction error results in a set of error regions for which shape or contours can be encoded by one or several methods, for instance chain or skeletons [8].

In block-based coders adapted to perform object scalability, the environment is quite different. The shape to be coded does not necessarily correspond to a uniform region, but rather to a preselected object from the scene, possibly including several subregions. Segmentation is performed semi-automatically based on semantic information (for instance the number of expected objects, as in [21]), or is available from the image composition stage (blue screening technique). Motion and texture coding remain block-based, but use shape masking to improve their performance on object boundaries [10, 36]. In this case, when lossy shape coding is performed, the reconstructed shape must be available to perform motion estimation (only pixels within the shape in the block will be considered). Shape temporal prediction must then use either former texture-based motion information, or specific motion vectors that need to be transmitted.

Considering the afore mentioned statements, the CAE technique can be straightforwardly extended [6] to the inter mode case to take advantage of the correlation between two successive masks. Given a motion compensated version of the previous mask, a modified template may include pixels from it, as shown in Figure 4. Similar steps have been applied to improve the MMR method as well [46].

Vertex-based methods can be adapted to provide inter frame coding without too much effort. In a block-based coder, prediction can be performed from the previous shape, by exploiting available motion information. This motion can be either specifically computed for shape information, thus requiring the corresponding parameters to be encoded, or extrapolated from the texture motoin vectors from the previous frame. Each vertex is then motion compensated and the corresponding error with regard to the contour to approximate is computed. If the error is lower than the predefined threshold, the vertex is retained, otherwise it is rejected. The retained vertices constitute a rough polygon to which the iterative refinement technique presented in section VI is applied. The vertex positions in the old and new lists of vertices are then encoded, for instance by means of a run-length coding method. Inserted vertices may also be encoded using adaptive arithmetic coding [12, 20].

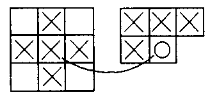

Fig 4. 9-pel template used for *inter* mode coding. The right side of the template represents pixels from the current mask and the left side pixels from the previous motion compensated mask. The arc defines the alignment of both masks.

Voisinage de 9 éléments utilisé pour le codage inter. *La partie de droite représente les éléments de l'image courante et celle de gauche les éléments de l'image précédente après compensation de mouvement. L'arc définit l'alignement entre les deux parties.*

Another possibility consists in coding the shape prediction error as a shape image. However, since the corresponding regions are expected to be small, the corresponding cost may be higher. Thresholding may be used to eliminate meaningless regions. The tolerance error should be adapted to a value different from that used in intra-coding. Similarly to motion/texture coding schemes, in case no matching is possible, for instance when no motion vector resulting in an acceptable prediction error can be found the shape is coded in intra mode.

VII.2. Entropy based prediction schemes

An alternative temporal prediction technique which does not use any motion information while remaining lossless may be used at the entropy coding stage. When sophisticated models are used to code the chain code symbols describing the shape, it is possible to exploit the experience gained from a previous frame to entropy code the current frame by maintaining the model from frame to frame, instead of resetting it after each frame.

As an illustration, in the context of chain coding using fixed-order, finite-context Markov models for arithmetic coding (see section V. 1.3), there exists a simple solution to exploit temporal redundancies to increase compression ratios. Instead of resetting the probability tables before each frame, the same tables can be used throughout the sequence. The cost of the first frame is the same as without exploiting temporal redundancies, but the cost of the subsequent frames decreases rapidly.

VIII. PERFORMANCE MEASURE

Bitrate evaluation for contour coding techniques is often provided in terms of the average number of bits per contour pixel, or per link as mentioned in section V. Such a measure depends on the contour definition, and the chosen connectivity. Comparaison with other, non contour based, techniques is therefore difficult. Furthermore, it does not show how many bits are actually necessary to encode the shape information of a given region, which may be critical in low bitrate applications. In the applications considered here, shape coding is not the final goal; its bitrate should be considered in the context of a complete shape-motion-texture coding scheme.

As is the case with the texture coding quality measure, a shape distortion measure is not easy to define objectively. The human visual system is very sensitive to edges, and errors in reconstructed shapes are far less acceptable than imperfect textures. If the shape is coded in a lossy fashion, imperfect boundary reconstruction may result in an annoying mixed background/foreground at the object borders. However, this argument is mainly valid for hybrid synthetic/natural image coding, where a perfect object mask can be obtained from computer-based object synthesis. Segmentation of natural images is not as perfect and usually provides noisy masks for which the use of lossy coding may be necessary, for instance by prefiltering the shape before coding.

An objective distortion measure requires the comparison between the original and reconstructed shapes. The comparison may be based on either the contour information, or the mismatched area. A contour-based error measure is more natural in contour coding methods, especially those minimizing an error criterion (geometrical representations). The distance from the original curve to the reconstructed curve needs then to be computed. Since no mathematical description of the original curve, quantized to a pixel-based 2D grid, is available, the error must be experimentally computed as a pixel-to-pixel distance. This allows an easy definition of the local error, such as the peak absolute deviation between two curves, but does not facilitate a global error measure. On the other hand, the number of mismatched pixels between the original and the reconstructed shape provides a global error measure, possibly scaled by the original shape size, but does not provide any local error information [30]. Yet the latter error measure offers the additional advantage that it is valid for any shape or contour representation method, and will therefore be used for performance comparison purpose in this paper.

IX. EXPERIMENTAL RESULTS

Simulations have been performed on a representative subset of MPEG-4 video test sequences:

- *Akiyo* sequence, speaker layer, QCIF (176×144 pixels per frame), 10 Hz. This sequence is a typical video-telephony sequence: smooth and temporally stable head-and-shoulders image.
- *Hall* sequence, right man layer, QCIF, 10 Hz. The region of interest is a man carrying a monitor in a hall way. The man walks towards the camera. The corresponding shape is small with sharp meaningful details and some tiny holes.
- *Children* sequence, logo layer, QCIF, 10 Hz. This is a logo sequence ("MPEG-4 world"), with translation and scaling. It contains many meaningful details.
- *Bream* sequence, fish layer, QCIF, 10 Hz. The object of interest is a computer generated swimming fish with a smooth shape. Motion is complex (shape deformation).

The following shape coding techniques have been investigated:

- Context-based arithmetic encoding, as per MPEG-4 verification model version 7.0 [36], that is a macroblock-based implementation including motion estimation and compensation.
- Differential chain code. A simple Huffman coding scheme is used to encode differential 8-connected links. This technique has been applied to original frames (lossless) and morphologically filtered shapes.
- Chain code with PPM entropy coding. This technique has been applied to original frames (lossless), morphologically filtered shapes, and quadtree-rounded shapes. To obtain the best possible results, the order of the Markov model has been fine tuned separately for each simulation.
- Hybrid macroblock/quadtree representation with different truncation parameters, as defined in the former MPEG-4 verification model version 2.0 [23].
- Polygonal approximation with fixed-length coding. Since no vertex entropy coding was used, further improvement can be obtained by applying differential and arithmetic coding to the vertex coordinates [20].
- Joint block/Hermite curve approximation. Iteration is stopped when a predefined number of control blocks is reached (40 in the presented results). The technique is not applied to *Hall* and *Children* because the corresponding shapes contain meaningful details smaller than the chosen block size (8×8 pixels).

The corresponding rate performances, measured as the averaged number of bits per frame for the sequence, are presented in Tables IV to VIII for different distortion constraints. Shape images are first cropped to their corresponding bounding boxes extended to multiples of 16 pixels, as defined by the MPEG-4 verification model [36]. Figure 5 to 11 plot the rate distribution on the *Bream* sequence for each technique. The bitrate values should be compared with the original non-compressed data size (one bit per pixel), as well as with the performance of a simple Lempel-Ziv entropy coding algorithm directly applied to the shape images (Unix compress command). The corresponding measures are provided in Table IX.

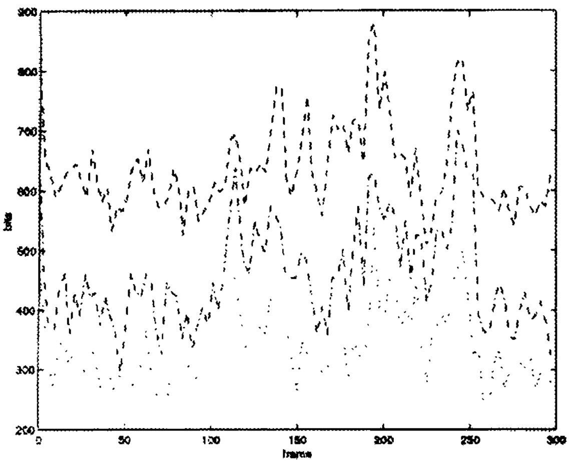

Fig 5. CAE method with temporal prediction, rate performance for lossless and lossy shape representation.

Méthode CAE avec prédiction temporelle, performance en compression pour des représentations sans et avec pertes.

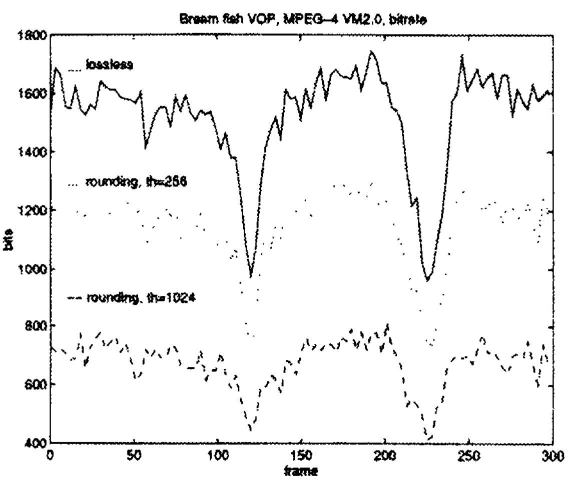

Fig 6. Hybrid quadtree/macroblock representation (MPEG-4 video VM2.0), rate performance for various error thresholds.

Performance en compression pour la technique hybride arbre quaternaire/macro-bloc (modèle de vérification MPEG-4 vidéo v2.0).

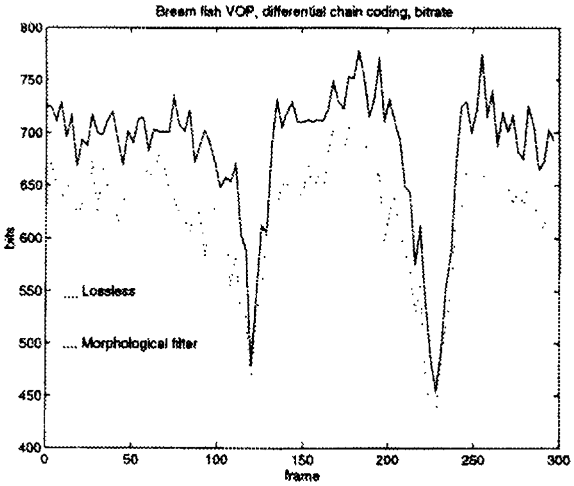

Fig 7. Differential chain coding, rate performance for lossless and pre-filtered shape representation.

Codage différentiel de chaînes, performance en compression pour des représentations sans pertes et après préfiltrage.

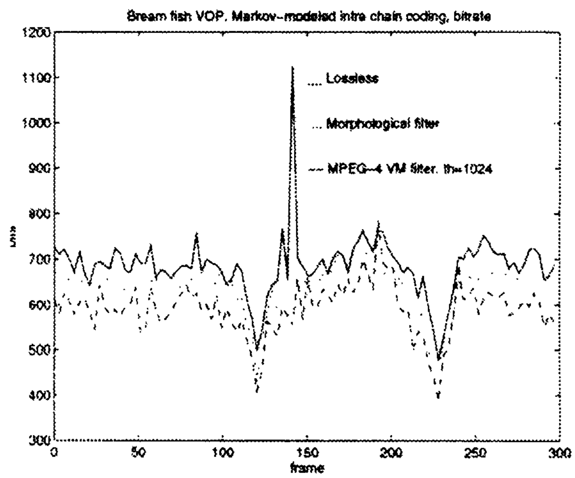

Fig 8. Markov-modeled intra chain coding, rate performance for lossless and pre-filtered shape representation.

Codage intra de chaînes par un modèle markovien, performance en compression pour des représentations sans pertes et après préfiltrage.

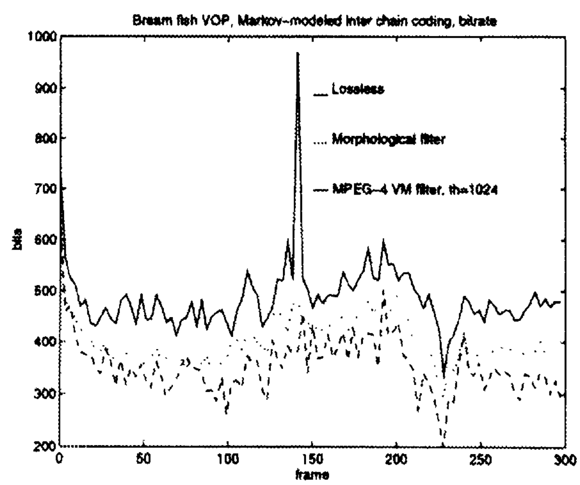

Fig 9. Markov-modeled inter chain coding, rate performance for lossless and pre-filtered shape representation and prediction.

Codage temporel de chaînes par un modèle markovien, performance en compression pour des représentations sans pertes et après préfiltrage.

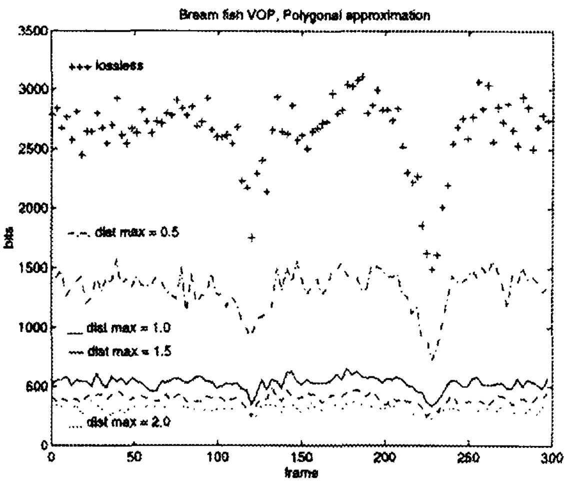

Fig 10. Polygonal approximation, rate performance for various error thresholds.

Approximation par polygones, performance en compression pour différents seuils d'erreur.

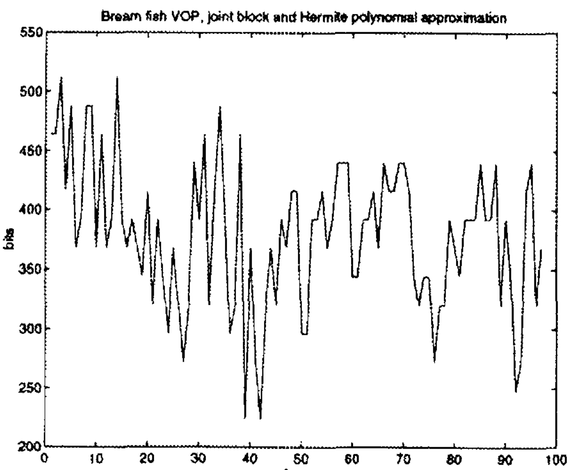

Fig 11. Joint block/Hermite polynomial approximation, rate performance, 40 control blocks.

Approximation conjointe par blocs et courbes de Hermite, performance en compression pour un nombre de blocs de contrôle fixé à 40.

Objective distortion measures are provided for different lossy schemes in Table X. The resulting distortion can be visually evaluated for the 102nd frame of the *Bream* sequence in Figure 12. The morphological filter appears to allow the most faithful reconstruction. However, it is not suited for shapes with small meaningful details, such as *Hall* and *Children*. The quadtree truncation process results in specific blocking artifacts. Since polygonal approximation splits the contour into straight lines, the reconstructed contour is not smooth. On the other hand, Hermite spline approximation based on control blocks preserves the contour smoothness, but results in a raw approximation of the shape due to its incapability to represent sharp corners (such as the tail of the fish). Although CAE uses a down-sampling / up-sampling scheme for efficient compression, the reconstructed shape is fairly smooth, thanks to a filter which is applied after up-sampling.

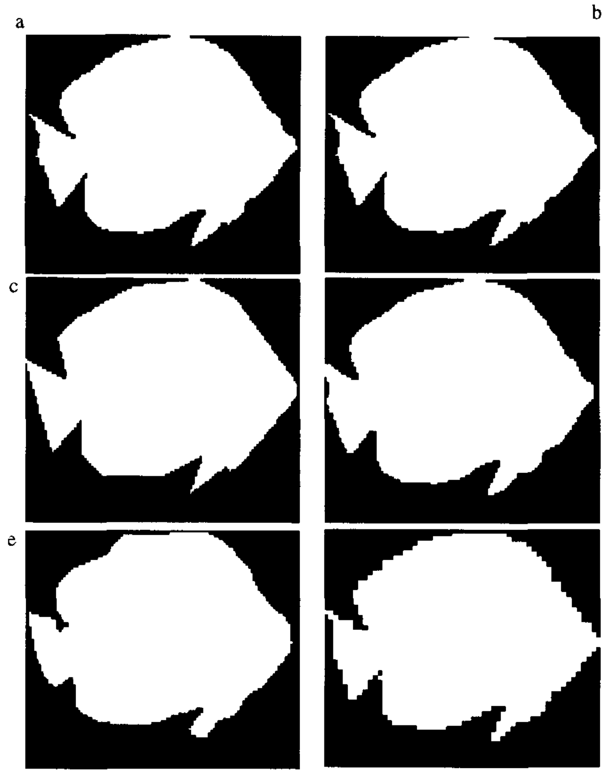

Fig 12. Comparison of the visual distortion yielded by different lossy schemes, *Bream* sequence, fish layer, frame ≠ 102.
(a) original QCIF bounding box (b) MPEG-4 truncation process
(c) polygonal approximation, d_{max} = 2.0 (d) morphological filter
(e) joint block/Hermite curve approximation
(f) Quadtree truncation process, *threshold* =1024.

Comparaison de la distorsion visuelle générée par différents modes de codage avec pertes, séquence Bream, *image ≠ 102. (a) rectangle englobant original en QCIF (b) méthode d'arrondi MPEG-4 (c) approximation par polygones, d_{max} = 2.0 (d) filtrage morphologique (e) approximation conjointe par blocs et courbes de Hermite (f) méthode d'arrondi de l'arbre quaternaire, seuil = 1024.*

At high threshold values the quadtree pre-processing method may also be exploited by PPM chain coding, which is able to model the specific right turns introduced by the blocking effect. However, this truncation process is not suited for the 8-connected differential chain code that associates low probabilities to the right angles. Since the lossy truncation pre-processing is grid-based, the resulting blocking artifact at the shape boundaries is visually annoying. In addition, it is not stable over time, and results in annoying blinking effect in the shape video sequence.

Block-based shape coding is generally more expensive than frame-based schemes. The lossless compression performance of the quadtree method is much lower than the chain coding performance, and the low bitrate obtained for the joint block/Hermite curve approximation method is reached at the cost of a relatively high distortion.

Table I: MPEG-4 video VM v7.0 performance (CAE method) without temporal prediction.
Performance du VM MPEG-4 v7.0 (méthode CAE) sans prédiction temporelle

rate (bits)	Lossless	Threshold = 32	Threshold = 64
Akiyo	601	518	380
Bream	785	660	494
Hall	276	263	198
Children	1924	1915	1641

Table II: MPEG-4 video VM v7.0 performance (CAE method) with temporal prediction.
Performance du VM MPEG-4 v7.0 (méthode CAE) avec prédiction temporelle

rate (bits)	Lossless	Threshold = 32	Threshold = 64
Akiyo	293	194	180
Bream	640	458	337
Hall	241	221	145
Children	561	493	376

Table III : Hybrid quadtree/macroblock (MPEG-4 video VM v2.0) performance.
Performance de la méthode hydride arbre quaternaire/macro-bloc (VM MPEG-4 v2.0)

rate (bits)	Lossless	THqt = 32	THqt = 64
Akiyo	1438	1216	714
Bream	1520	1147	675
Hall	425	221	184
Children	3093	2487	1379

Table IV: Differential chain coding performance.
Performance du codage de chaînes différentiel.

rate (bits)	Lossless	Morphological
Akiyo	679	660
Bream	685	623
Hall	274	217
Children	2390	1660

Table V: Markov-modeled intra chain performance.
Performance du codage intra de chaînes avec modèle markovien.

rate (bits)	Lossless	Morphological	VM2.0 Thqt= 1024
Akiyo	516	450	424
Bream	685	639	591
Hall	300	188	239
Children	2255	1651	2176

Table VI: Markov-modeled inter chain coding performance.
Performance du codage temporel de chaînes avec modèle markovien.

rate (bits)	Lossless	Morphological	VM2.0 Tqt = 1024
Akiyo	309	253	209
Bream	485	404	354
Hall	245	122	158
Children	1098	755	1046

Table VII: Polygonal approximation performance.
Performance de l'approximation par polygones.

rate (bits)	Lossless	dist max = 0.5	dist max = 1.0	dist max = 2.0
Akiyo	2210	1221	412	190
Bream	2639	1326	530	309
Hall	725	482	256	153
Children	8220	5657	3016	2197

Table VIII : Joint block/Hermite curve representation performance.
Performance de la représentation conjointe par blocs et courbes de Hermite.

rate (bits)	max control blocks 40
Akiyo	386
Bream	379

Table IX: Original and Lempel-Ziv compressed shape bounding boxes size.
Tailles des rectangles englobants originaux et compressés par l'algorithme de Lempel-Ziv.

rate (bits)	original bounding box	UNIX compressed
Akiyo	18437	5300
Bream	12484	4783
Hall	1627	no gain
Children	6607	5874

Table X: Distortion results for various lossy schemes.
Mesures de distorsion pour différentes méthodes avec pertes.

Distortion	AKIYO	BREAM	HALL	CHILDREN
Morphological	0.0004	0.0047	0.0588	0.2393
Quadtree th = 256	0.0038	0.0096	0.0346	0.075
Quadtree th = 1024	0.0224	0.0307	0.1465	0.3181
pol. dist = 0.5	0.0014	0.0035	0.0119	0.0635
pol. dist = 1.0	0.0072	0.0132	0.0657	0.1881
pol. dist = 2.0	0.0179	0.0279	0.1433	0.3053
block/Hermite	0.0386	0.055	not av.	not av.

In the case of polygonal approximation, it should be noted that when the tolerance distance increases the bitrate decreases, but the corresponding gain becomes less important at high error values (flat part of the rate / distortion curve). For the *Children* logo, the coding cost to represent a lot of short, highly curved contours is so important that the final lossless bitrate is higher than the original shape size. In such a case, the overhead required by a contour representation becomes very important, and also explains the relatively poor performance of chain code methods on such shapes.

X. GENERAL DISCUSSION

Several shape representation and coding methods have been presented. In this paper, bitmap, intrinsic shape and contour representation have been distinguished. Compression results for several test sequences have been shown in section IX.

For most applications, high compression rates are desirable, since storage space and bandwidth are often at a premium. For low-bandwidth applications, such as videotelophony, the goal is to utilize as much as possible of the bandwidth that is available. Rate-distortion control schemes are thus needed to get the lowest distortion for a given rate. For multimedia client-server applications, scalability is desirable. After a few bits have been transmitted, the receiver should be able to render a coarse approximation of the shape, which is then refined as more bits are received. For networking applications, error resilience is desirable. The loss or damaging of a packet should not completely ruin the reconstruction of the shape.

In the framework of MPEG-4, the bitmap-based method comes out to be the most efficient in terms of compression. This superiority is mainly due to the predictive coding, which takes advantage of the temporal redundancy. These methods are generally quite versatile and can be adapted to either frame-based or block-based coding environments. Also adaptations to scalable [5] and error-resilient [7] environments have been proposed.

Skeleton decomposition is a well known way to intrinsically represent shapes, and has recently been applied to lossless shape coding. However, the corresponding performance remains lower than that of chain code based methods [8], especially when the shape can be described by a few long contours. On the other hand, the skeleton decomposition allows scalable transmission and can offer a graceful degradation when transmission errors occur by allowing at least a partial reconstruction of the shape.

A hybrid macroblock/quadtree representation is well suited for low-delay applications since each macroblock is coded independently. This also facilitates error resilience. Progressive transmission can be obtained by exploiting the quadtree hierarchical representation. However, the corresponding compression performance remains poor. A simple rate/distortion control based on a specific pre-processing step can be used, but the resulting image degradation is not graceful.

Contour oriented coding schemes provide a semantic representation suitable for some applications such as indexing and retrieval. However, their efficiency decreases when the shape is a collection of small disjoint regions, and their integration in a block-based coder is not straightforward.

The chain coding method, possibly improved by highly efficient entropy coding models, provides the most efficient contour-based lossless compression scheme. However, current schemes have poor (or no) rate distortion control, poor scability, and poor (or no) error robustness. Further research is needed in order to make the chain coding method more versatile, possibly by combining it with geometrical contour approximation.

The main strength of geometrical approximation methods lies in their capacity to gracefully degrade the shape quality as the maximum allowable error constraint is relaxed, thus easily providing a rate/distortion control. Lossless coding may be achieved by setting this error to zero; however, most shapes are not simple enough for this technique to outperform the chain coding methods in that case. On the other hand, such methods may be very useful for representing noisy masks, since they naturally filter the shape. Progressive transmission can also be achieved by transmitting vertices in a specific order allowing shape reconstruction from the coarsest to the finest description. The error resilience is slightly better than for coding, depending on whether absolute or differential coding of the vertices is performed.

In the framework of video communications, the performance of shape coding schemes can be improved by taking temporal redundancies into account as well. More research is needed in this area. Basically, any intra shape coding technique may be extended to handle temporal prediction. One approach, specifically suitable for lossless chain code methods, consists in exploiting temporal redundancy at the entropy coding level. Another approach, expected to be more efficient uses motion estimation and compensation techniques to predict shape changes from frame to frame.

In conclusion, a lot of research still needs to be done in the field of coding the shape of video objects. Future possible solutions may lie in hybrid coding schemes mixing different shape representation methods, such as a scalable chain coding method as extension of polygonal approximation, or the use of a semantic shape representation in intra frames for high-level random access combined with an efficient block-based temporal coding method. Moreover, when designing any shape coding scheme, attention should be paid not only compression efficiency, but also to interaction with texture coding, to rate-distortion control capability and additional functionalities such as error robustness and scalability.

Manuscrit reçu le 22 décembre 1997

REFERENCES

[1] ANSARI (N.), DELP. (E. J.), On detecting dominant points. *Pattern Recognition,* 24 :441-450, (1991).

[2] AYER (S.), Sequential and competive methods for estimation of multiple motions. *PhD thesis*, EPFL, Lausanne, Switzerland, (1995).

[3] BALLARD (D.), BROWN (C.), Computer vision. *Prentice Hall*, 1982.

[4] BOSSEN (F.), EBRAHIMI (T.), Region shape coding. *Technical Report* M0318, ISO / IEC JTC1 / SC29 / WG11, (Nov. 1995).

[5] BOSSEN (F.), EBRAHIMI (T.). A simple efficient binary shape coding technique based on bitmap representation. *Int. Conf. on Acoustics, speech and signal processing*, (1997).

[6] BRADY (N.), BOSSEN (F.), MURPHY (N.). Context-based arithmetic encoding of 2D shapes sequences. *Int. Conf. on Image Processing*, (1997).

[7] BRADY (N.), DUCLA-SOARES (L.). Error resilience of arbitrarily shape vos (ce e14). *Technical Report* M2370, ISO / IEC JTC1 / SC29 / WG11, (July 1997).

[8] BRIGGER (P.). Morphological shape representation using the skeleton decomposition : application to image coding. *PhD thesis*, EPFL, Lausanne, Switzerland, (1995).

[9] BRIGGER (P.), Kunt (M.). Morphological contour coding using distance functions optimized by genetic algorithms. *Int. Conf. on Image Processing*, volume I, pp. 534-537, (Nov. 1995).

[10] LE BUHAN (C.), REUSENS (E.), EBRAHIMI (T.) . Object-scalable dynamic coding of visual information. *Proc. EUSIPCO*. (Sept 1996).

[11] CLEARY (J.), WITTEN (I.). Data compression using adaptive coding and partial string matching. *IEEE Trans. COM*-**32**(4), (Apr. 1984).

[12] BERLIN (HHI) COST211ter. Segmentation based coding scheme for object scalable and quality scalable coding of video. *Technical Report* M0415, ISO / IEC JTC1 / WG11, (Nov. 1995).

[13] DUDA (R. D.), HART (P. E.). Pattern classification and scene analysis. *Wiley*, New York, (1973).

[14] DUNHAM (J.). Optimum uniform piecewise linear approximation of planar curves. *IEEE Trans. PAMI*, **8**(1):67-75, (Jan. 1986).

[15] EDEN (M.), KOCHER (M.). On the performance of a contour coding algorithm in the context of image coding Part I: contour segment coding. *Signal Processing*, **8**:381-386, (1985).

[16] ETOH (M.), BOON (C.S.), and KADONO (S.). Template-based video coding with opacity representation. *IEEE Trans. CSVT.*, **7**(1):172-180, (Feb. 1997).

[17] FOLEY (J. D.), DAM (A.van), FEINER (S. K.), HUGHES (J. F.). Computer graphics : principles and practice. *Adision-Wesley*, 2nd edition, (1990).

[18] FREEMAN (H.). On the encoding of arbitrary geometric configurations. *IRE Trans. EC*-**10**: 260-268, (June 1961).

[19] FREEMAN (H.) Computer processing of line drawing images. *Computing Surveys*, **6**: 57-97, (March 1974).

[20] GERKEN (P.). Object-based analysis-synthesis coding of image sequences at very low bit rates. *IEEE Trans. CSVT*, **4**(3):-235, (1994).

[21] GU (C.). Multivalued morphology and segmentation-based coding. *PhD thesis*, EPFL, Lausanne, Switzerland, (1996).

[22] GU (C.), KUNT (M.). Contour simplification by a new nonlinear filter region-based coding. *IEEE International Conference on Image Processing, Los Alamitos*, volume 3, pp. 751-755, (1994).

[23] *** Ad hoc group on MPEG-4 video VM editing. MPEG-4 video verification model version 2.0. *Technical Report* JTC1 / SC29/ WG11 / N1260, ISO / IEC, (March 1996).

[24] *** Ad hoc group on shape coding. Description of core experiments on shape coding. *Technical Report* JTC1 / SC29 / WG11 / M0889, ISO / IEC, (May 1996).

[25] *** Recommendation T. 6. *Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus* ITU-T.

[26] *** Recommendation T. 82. *Information technology - coded representation of picture and audio information - progressive bilevel image compression* ITU-T.

[27] Kaneko (T.), Okudaira (M). Encoding of arbitrary curves based on chain code representation. *IEEE Trans. COM.*,**33**(7): 697-707, (July 1985).

[28] KUNT (M.), IKONOMOPOULOS (A.), KOCHER (M.). Second generation image coding techniques *Proc. IEEE*, **73**, (4): 549-575, (1985.)

[29] LANGDON (G.), RISSANEN (J.). Compression of black-white images with arithmetic coding. *IEEE Trans. COM*, **29**(6): 858-867, (June 1981).

[30] LI (H.), GERKEN (P.). Comparison of different error criteria for shape coding. *Technical Report* M0697, ISO / IEC JTC1 / SC29/ WG11, (March 1996).

[31] LONCARIC (S.), DHAWAN (A.). Near-optimal MST-based shape description using genetic algorithm. *Pattern Recognition*, **28**(4):571-579,(Apr. 1995).

[32] LU (C.) and DUNHAM (J.). Highly efficient coding schemes for contour lines based on chain code representations. *IEEE Trans. COM*, **39**(10): 1511-1514, (Oct. 1991).

[33] MARAGOS (P. A.), SCHAFER (R. W.). Morphological skeleton and coding of binary images.*IEEE Trans ASSP* **34**(5): 1228-1244, (Oct. 1986) :

[34] MEHROTRA (R.), GARY (J. E.). Similar shape retrieval in shape data management. *Computer*, **28**(9): 57-62, (1995).

[35 MOCCAGATTA (I.), SCHUTZ (M.), MOSCHENI (F.), DUFAUX (F.). Vector quantization (VQ)-based motion field refinement technique for image sequence coding. *Proc. SPIE - The International Society for Optical Engineering*, volume 2451, pp. 156-167. Society of Photo-Optical Instrumentation Engineers. Bellingham (1995).

[36] *** MPEG. Video verification model version 7.0. *Technical Report Available from* http : // drogo.cselt.stet.it / mpeg, ISO / IEC JTC1 / SC29 / WG11, (1997).

[37] RAMER (U.). An iterative procedure for polygonal approximation of plane curves. *Computer Graphics and Image Processing*, **1**: 244-256, (1972).

[38] RAY (B.), RAY (K.). A new split-and-merge for polygonal approximation of chain coded curves. *Pattern Recognition Letters*, **16**: 161-169, (Feb. 1995).

[39] RAY (B.), RAY (K.). An optimal algorithm for polygonal approximation of digitized curves using l. 1 norm. *Pattern Recognition*, **26**: 505-509, (Feb. 1995).

[40] RISSANEN (J.). Generalized Kraft inequality and arithmetic coding. *IBM Journal Res. and Dev.*, **20**: 198-203, (May 1976).

[41] ROSENFELD (A.), KAK (A. C.). Digital picture processing, volume 2. *Academic Press*, New York, USA, 2nd edition, (1982).

[42] SALEMBIER (P.), MARQUÉS (F.), GASULL (A.). Coding of partition sequences. *In* L. Torres and M. Kunt, editors, *Video coding. Kluwer Academic*, (1996).

[43] SHAPIRO (L.), MACDONALD (R.), STERNBERG (S.). Ordered structural shape matching with primitive extraction by mathematical morphology. *Pattern Recognition*, **20**: 75-90, (1987).

[44] STUHLMULLER(K. W.), SALAI (A.), GIROD (B.). Rate constrained contour representation for region-based motion compensation. *Proc. Symp. on Visual Comm. and image Proc., SPIE*, volume 2727, (March 1996).

[45] WRITTEN (I.), NEAl (R.), CLEARY (J.). Arithmetic coding for data compression. *Communications ACM*, **30**(6): 520-540, (June 1997).

[46] YAMAGUCHI (N.), IDA (T.), WATANABE (T.). A binary shape coding method using modified MMR. *Int. Conf. on Image Processing*, (Oct 1997).